Ethically Hacking an E-Commerce Website
Capstone Project

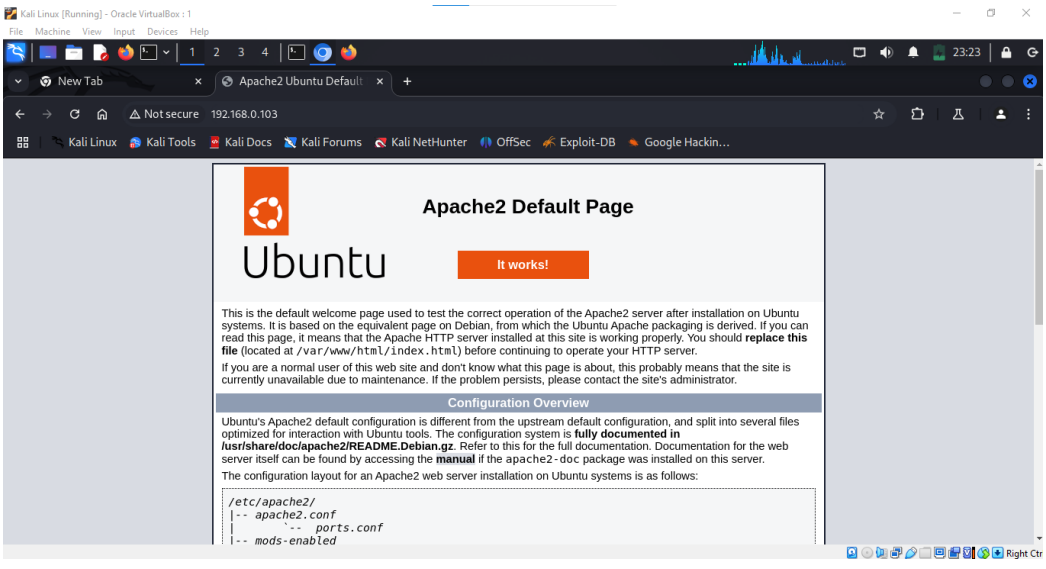
Title:
Cross-Site Scripting (XSS) in the Search Function of OWASP Juice Shop.

Description:
A Cross-Site Scripting (XSS) vulnerability was discovered in the search function of the OWASP Juice Shop application. The vulnerability allows an attacker to inject malicious JavaScript code, which executes in the victim's browser when they interact with the compromised page.

This vulnerability exists because the application fails to properly sanitize user input before rendering it on the page. As a result, an attacker can execute scripts that could steal user session cookies, perform phishing attacks, or even deface the website.

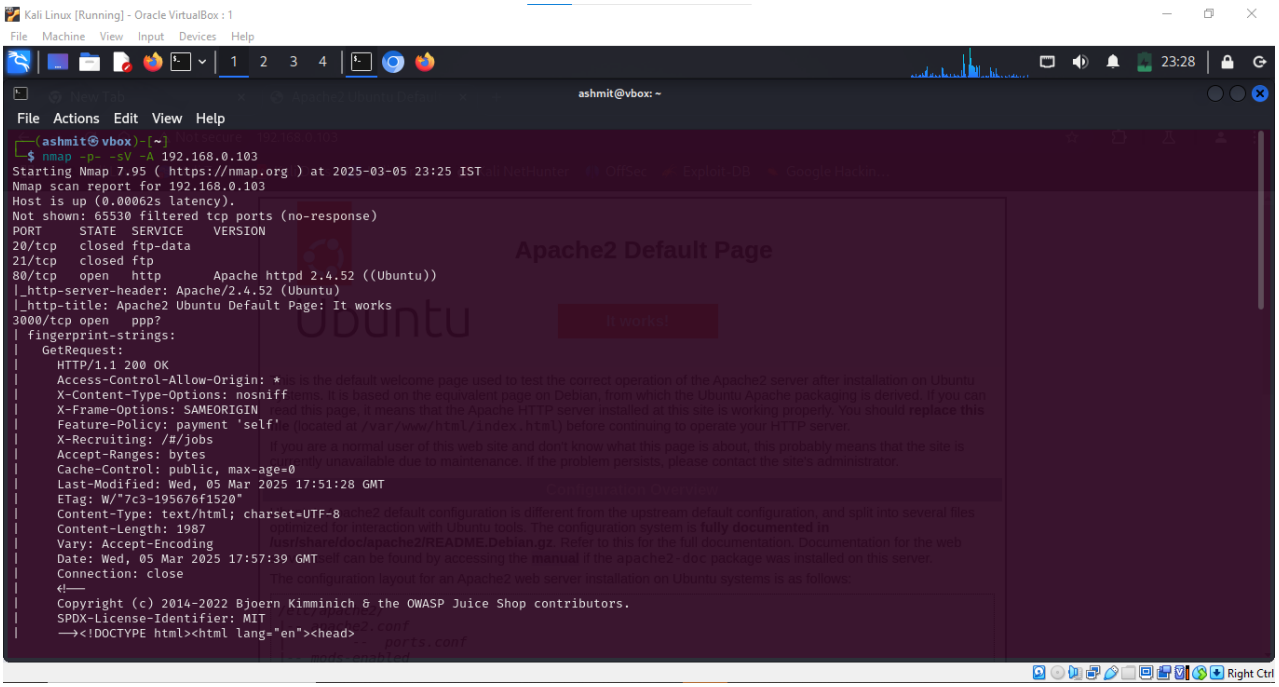
Steps to reproduce:
Step 1: Deploy the Virtual Machine:
1.Download the Sandbox E-Commerce site virtual box from the provided link:
- Sandbox E-Commerce site Virtual Machine

2.Open Oracle VirtualBox.
3.Click File → Import Appliance.
4.Select the downloaded Sandbox E-Commerce site challenge file and click Next.
5.Change MAC Address Policy to "Include all network adapter MAC addresses" and click Import.
6.After the import completes, go to Settings → Network and set Attached to: Bridged Adapter.
7.Click Start to deploy the machine.
8.Once deployed, the machine's IP address will be displayed.
The IP is 192.168.0.103



Step 2: Perform Reconnaissance

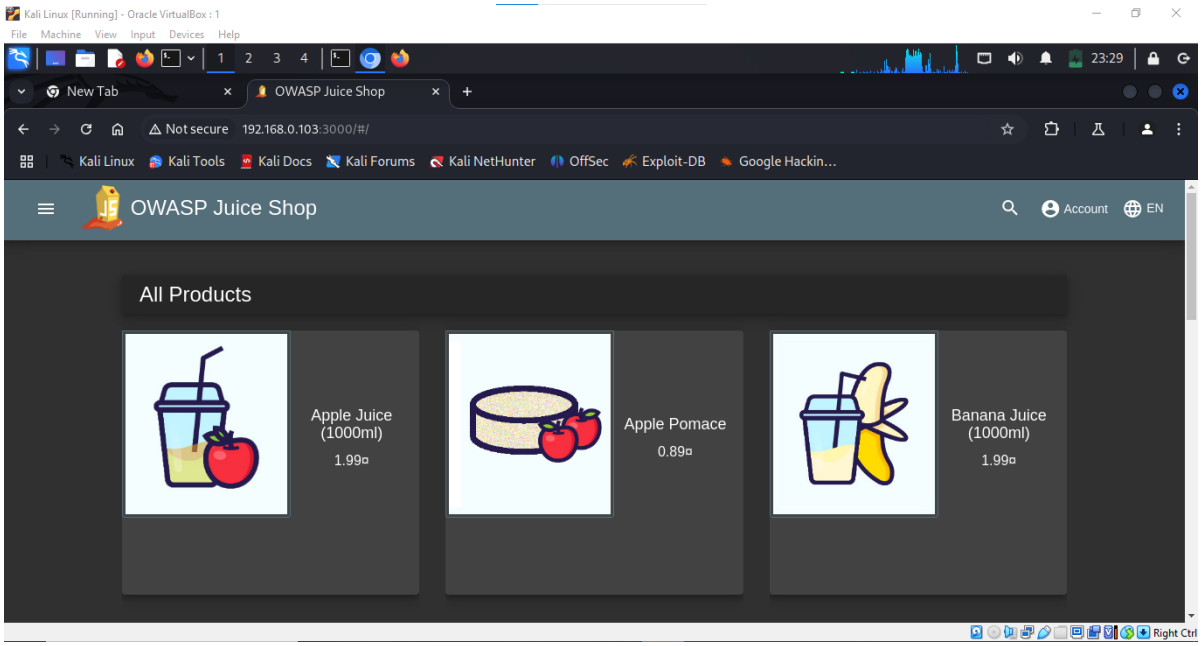
1. Find open ports and services.



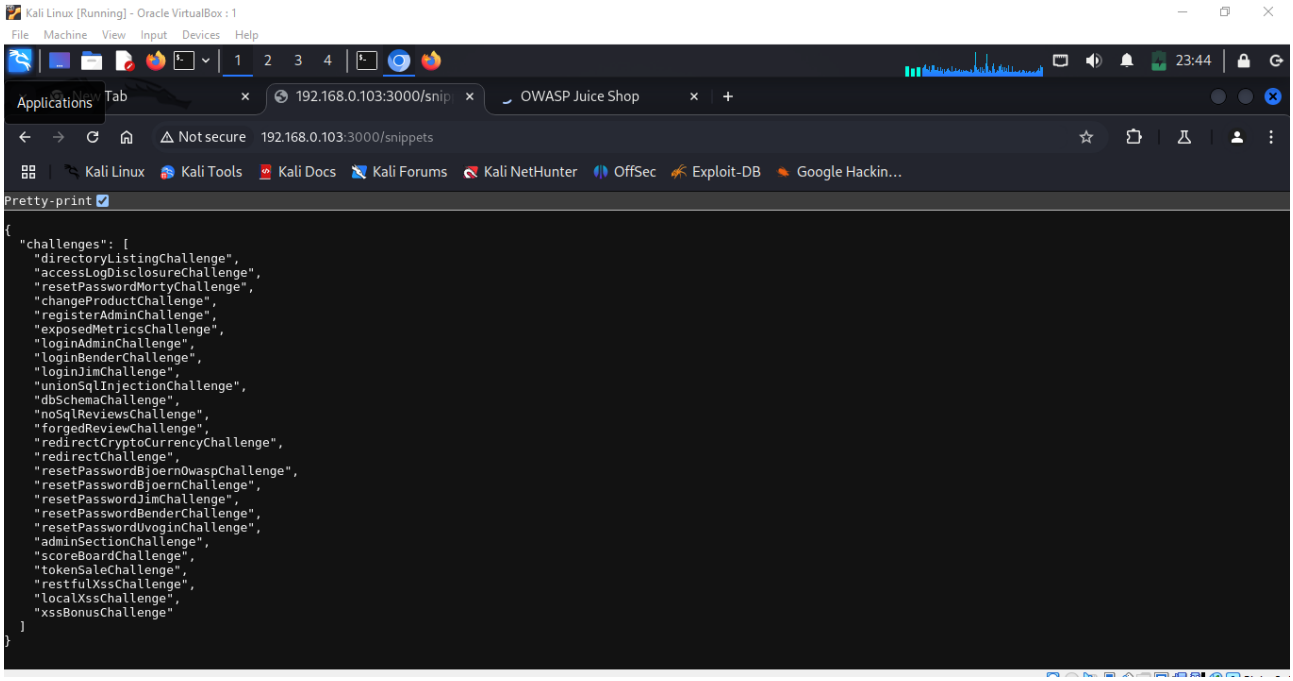
2. This will provide a list of open ports and services running on the machine.
3. Check for web services

Step 3: Exploit Vulnerabilities

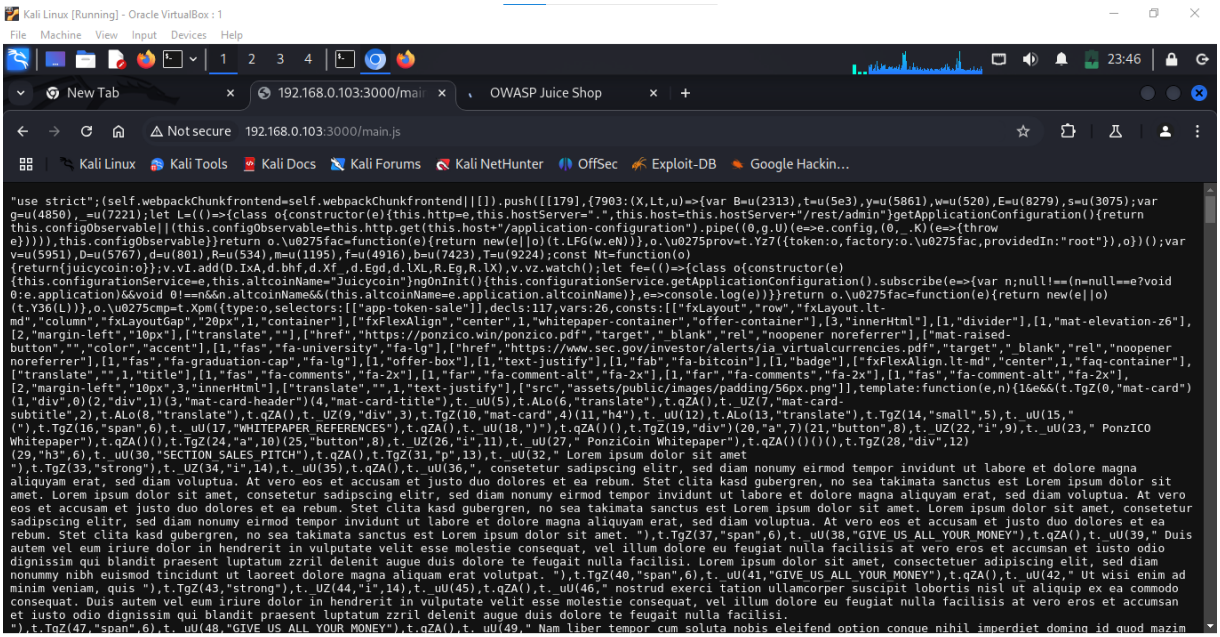
1. Found HTTP port is open.
2. Visit “http://192.168.0.103:3000/#/"



3. Visit “http://192.168.0.103:3000/snippets/"

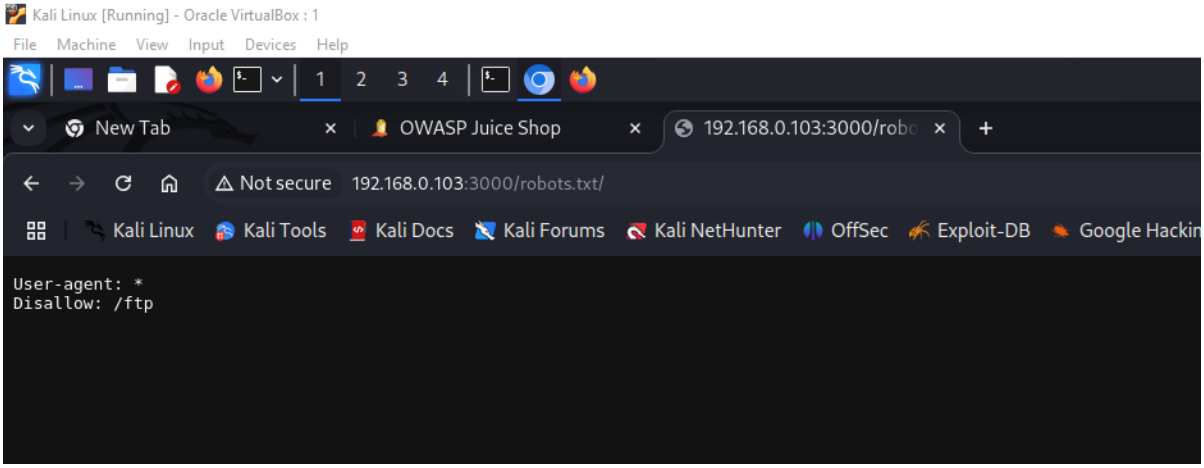


4. Visit “http://192.168.0.103:3000/main.js/"

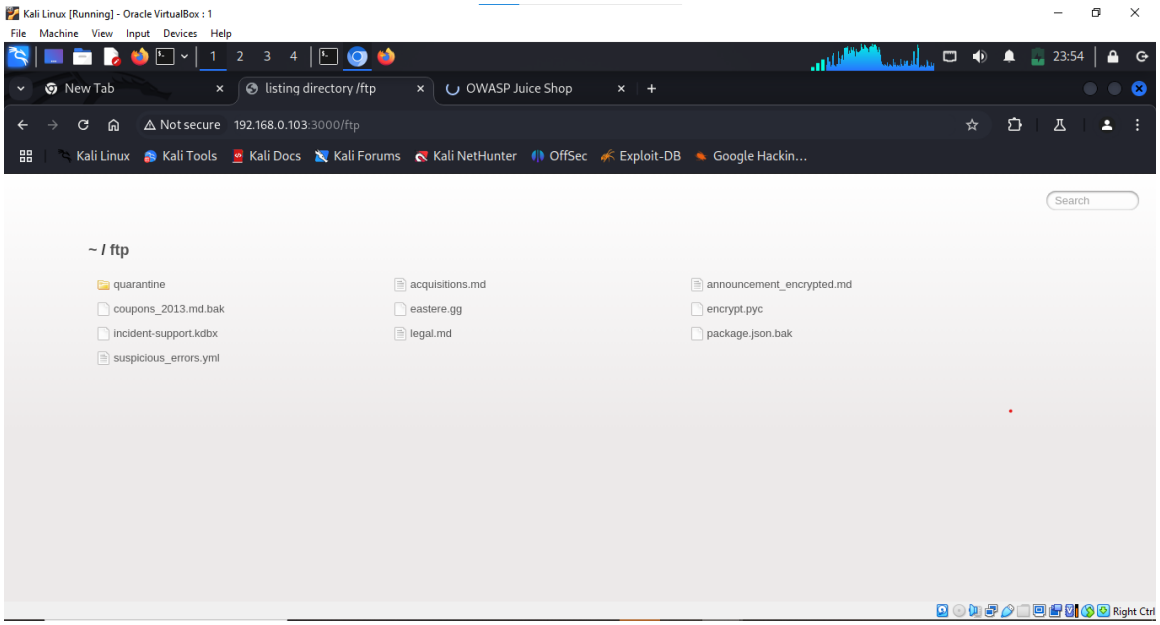


I. Directory Listing Challenge

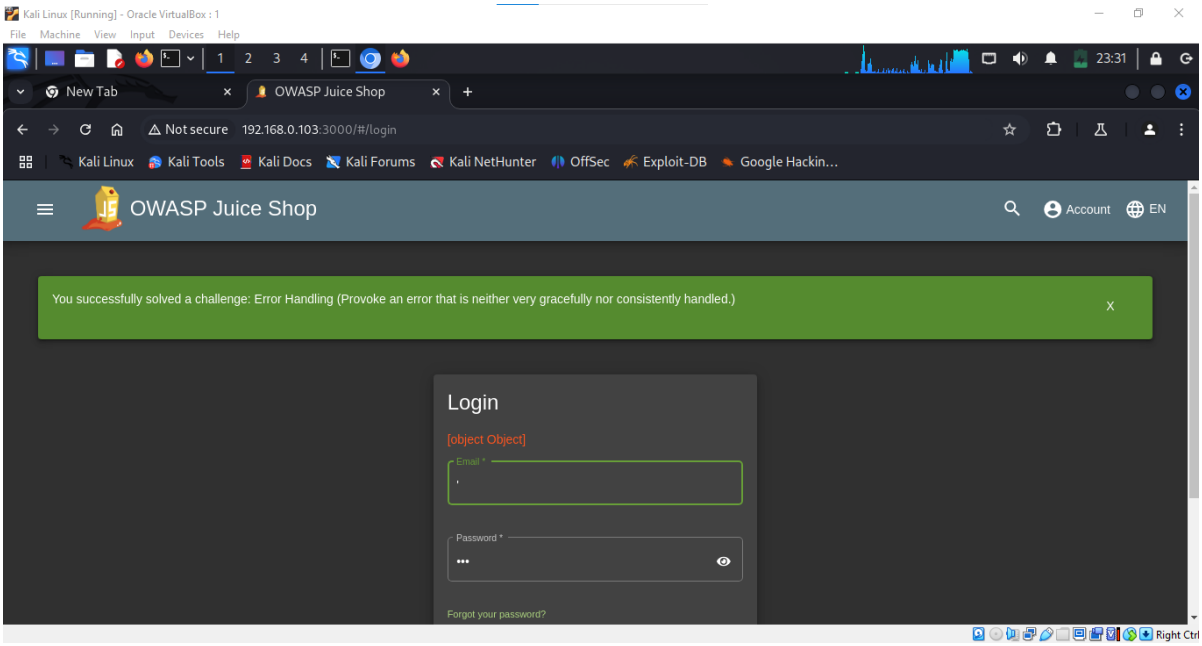
1. Visit “http://192.168.0.103:3000/robots.txt/"



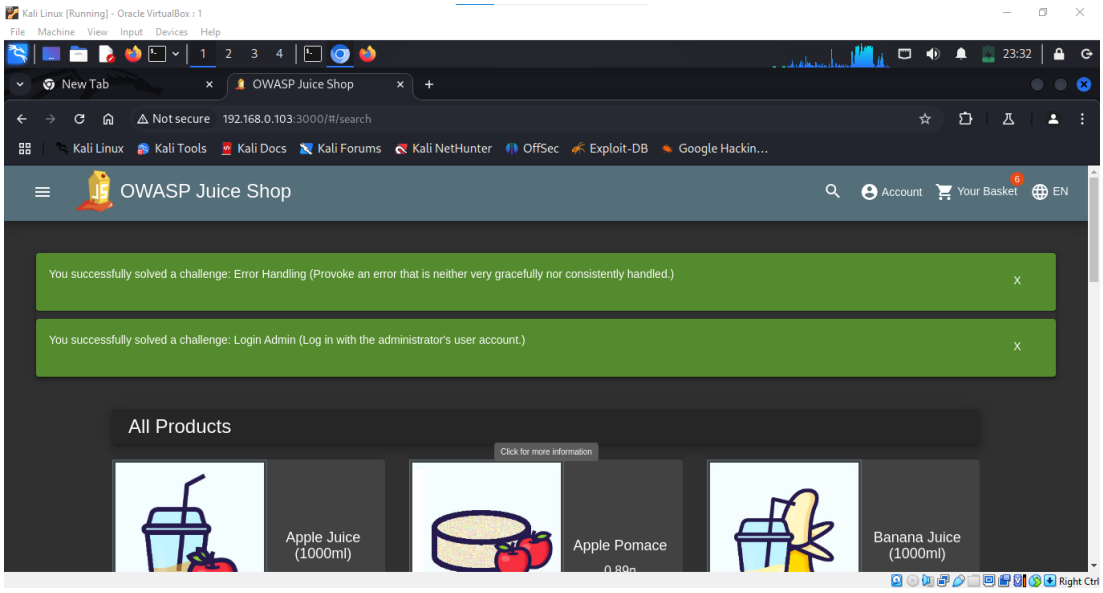
2. Visit "http://192.168.0.103:3000/ftp"



II. Login Admin

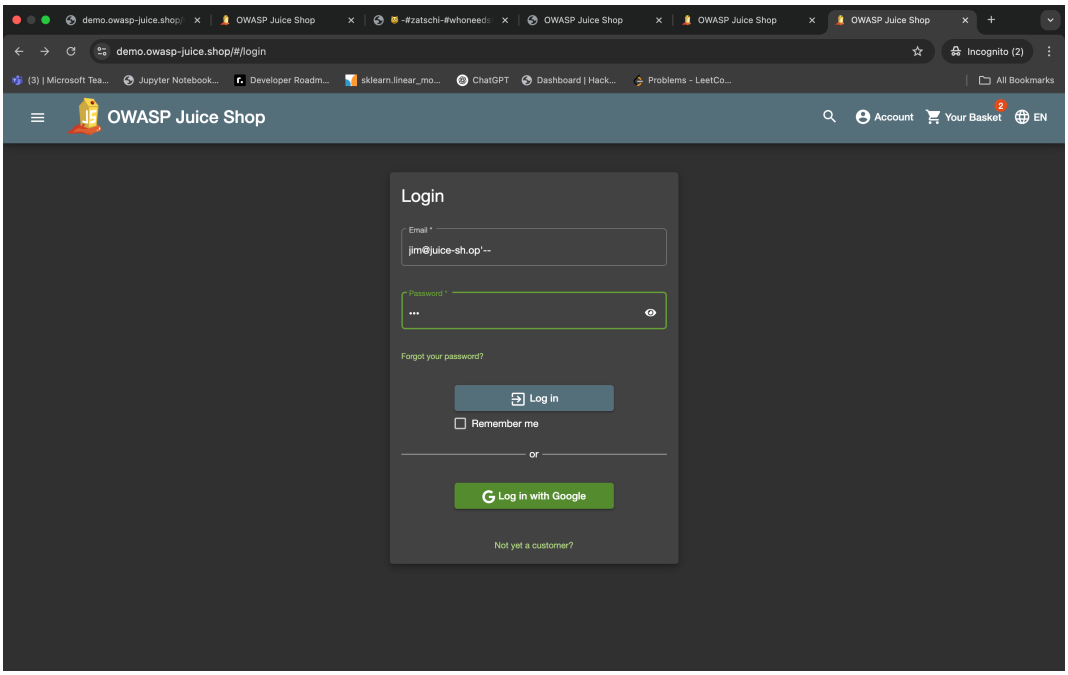


1. Log in with *Email* ' or 1=1-- and any *Password* which will authenticate the first entry in the Users table which coincidentally happens to be the administrator



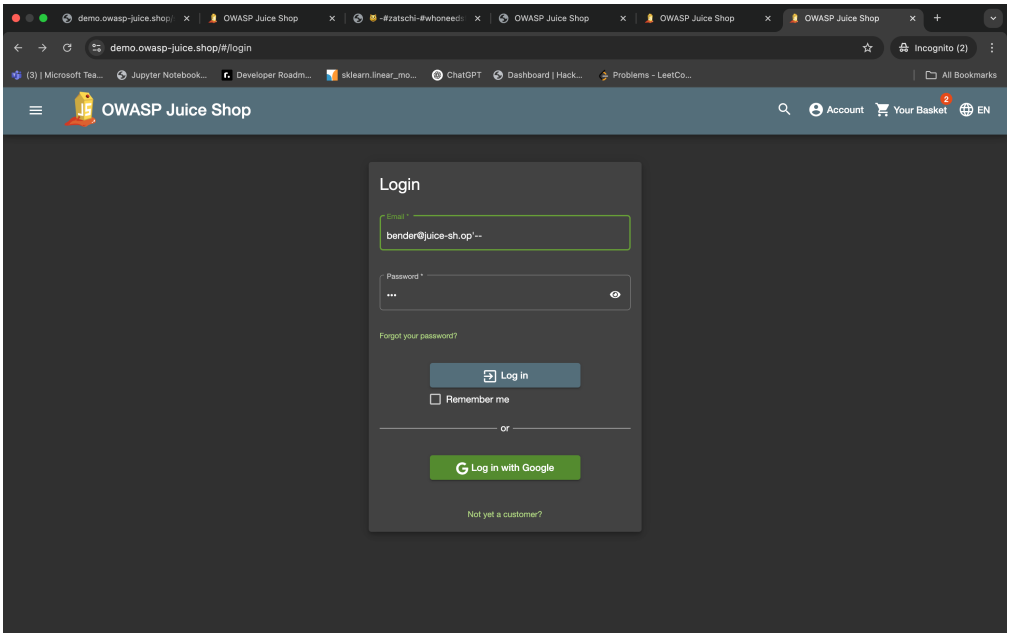
III. Log in with Jim's user account

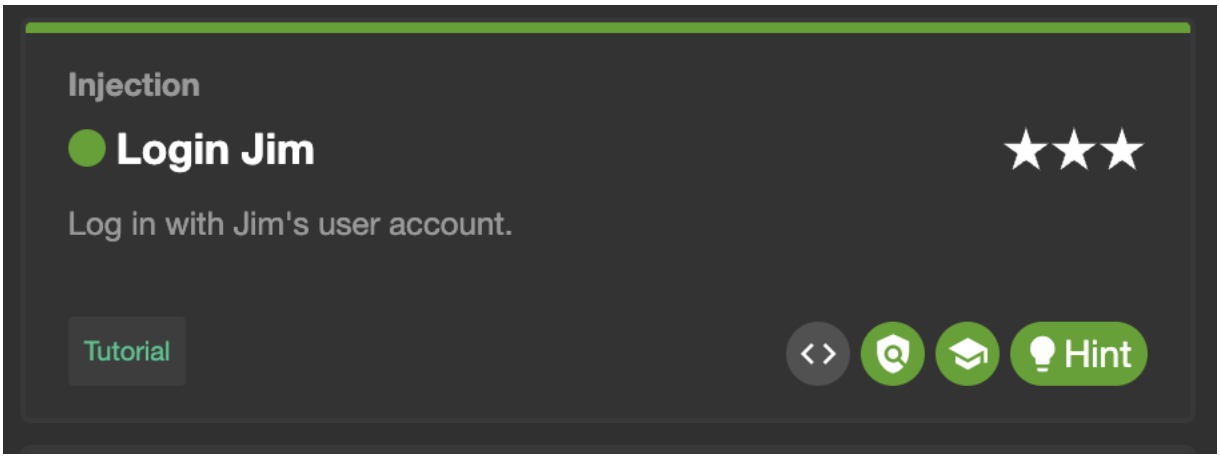
Log in with *Email* jim@juice-sh.op'-- and any *Password* if you already know the email address of Jim.

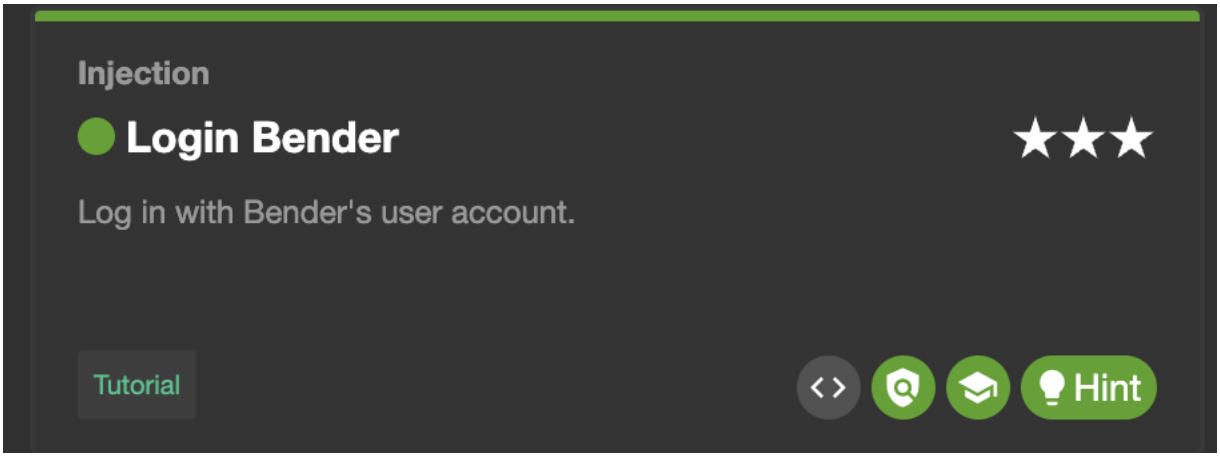


IV. Log in with Bender's user account

Log in with *Email* bender@juice-sh.op'-- and any *Password* if you already know the email address of Bender.



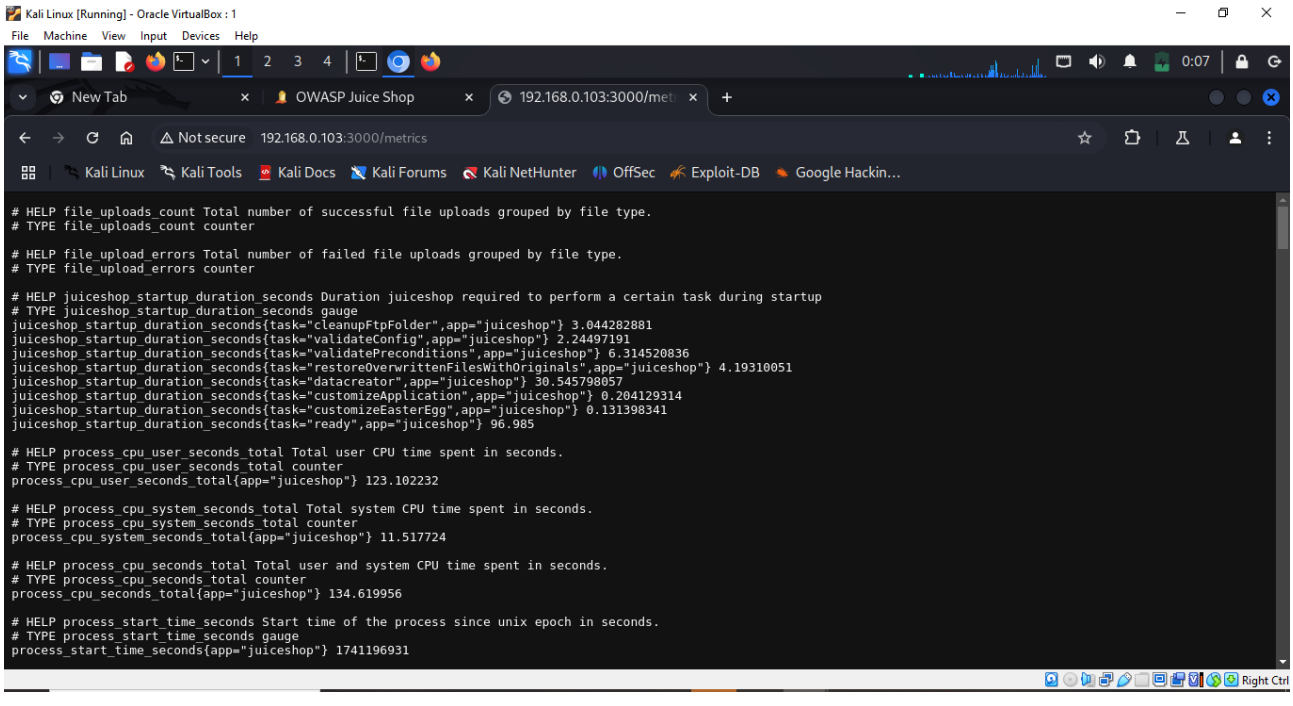




•

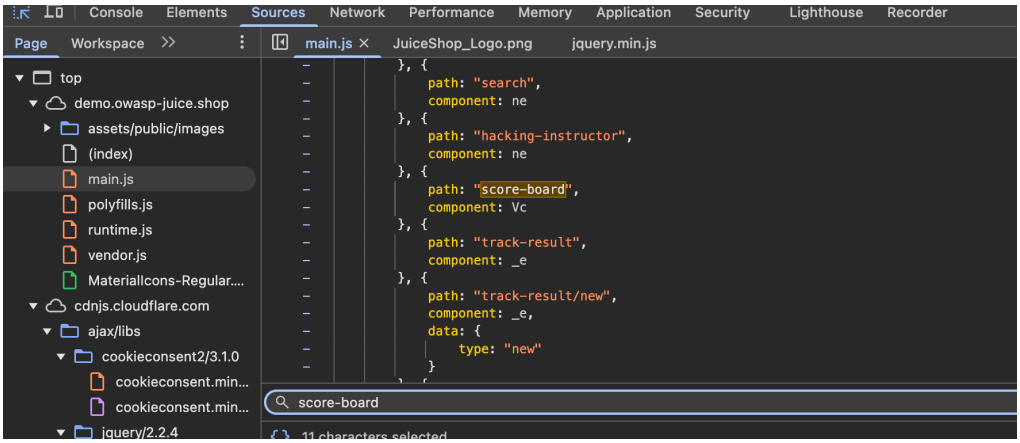
V. Exposed Metrics

1. Visit http://192.168.0.103:3000/metrics to view the actual Prometheus metrics of the Juice Shop and solve this challenge

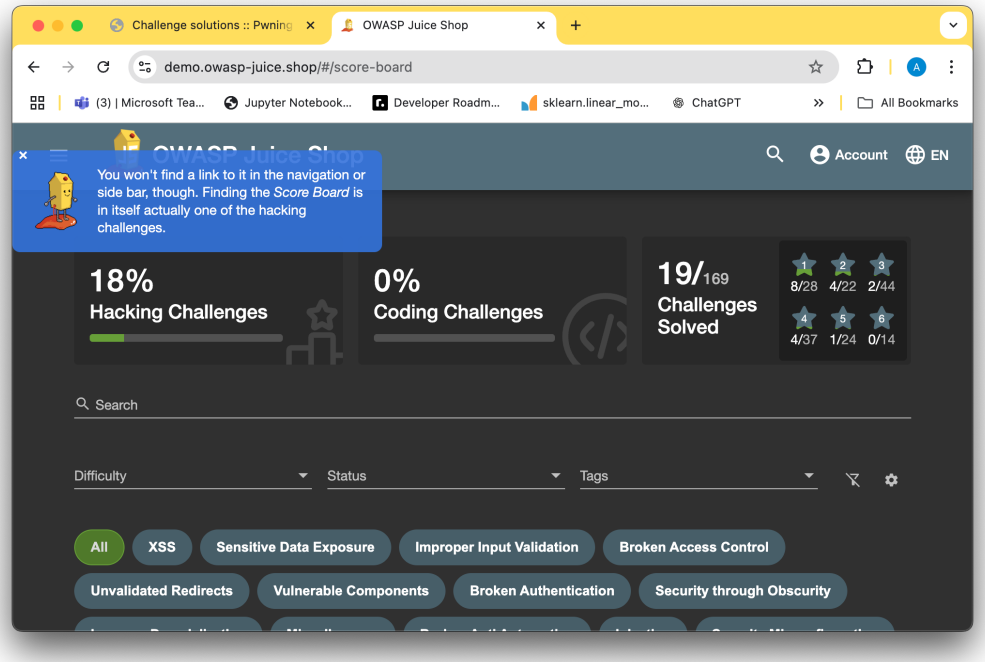


VI. Score Board

1. Go to the *Sources* tab of your browsers DevTools and open the main.js file.
2. If your browser offers pretty-printing of this minified messy code, best use this offer. In Chrome this can be done with the "{}"-button.
3. Search for score and iterate through each finding to come across one looking like a route mapping section:

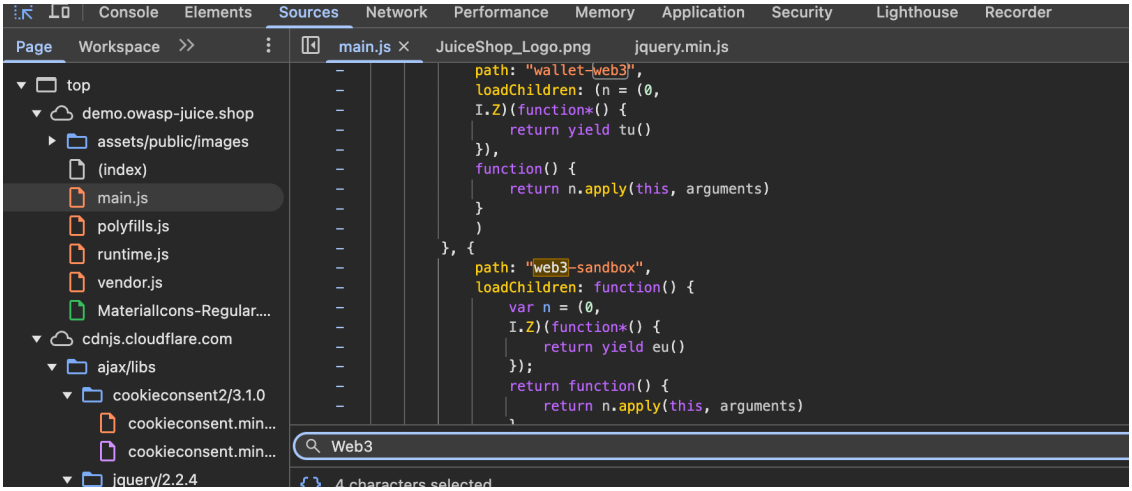


4. Navigate to http://192.168.0.103:3000/#/score-board to solve the challenge.
5. From now on you will see the additional menu item *Score Board* in the navigation bar.

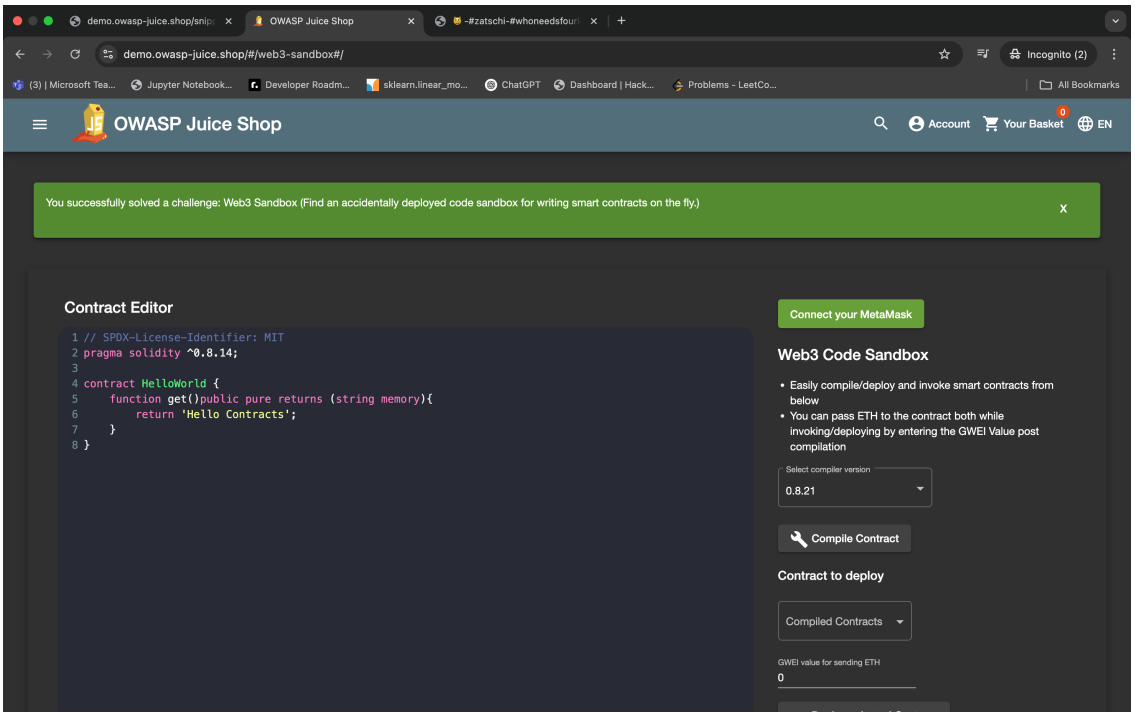


VII.Web3 Sandbox

1. Go to the *Sources* tab of your browsers DevTools and open the main.js file.
2. If your browser offers pretty-printing of this minified messy code, best use this offer. In Chrome this can be done with the "{}"-button.
3. Search for web3 or sandbox and iterate through each finding to come across one looking like a route mapping section:

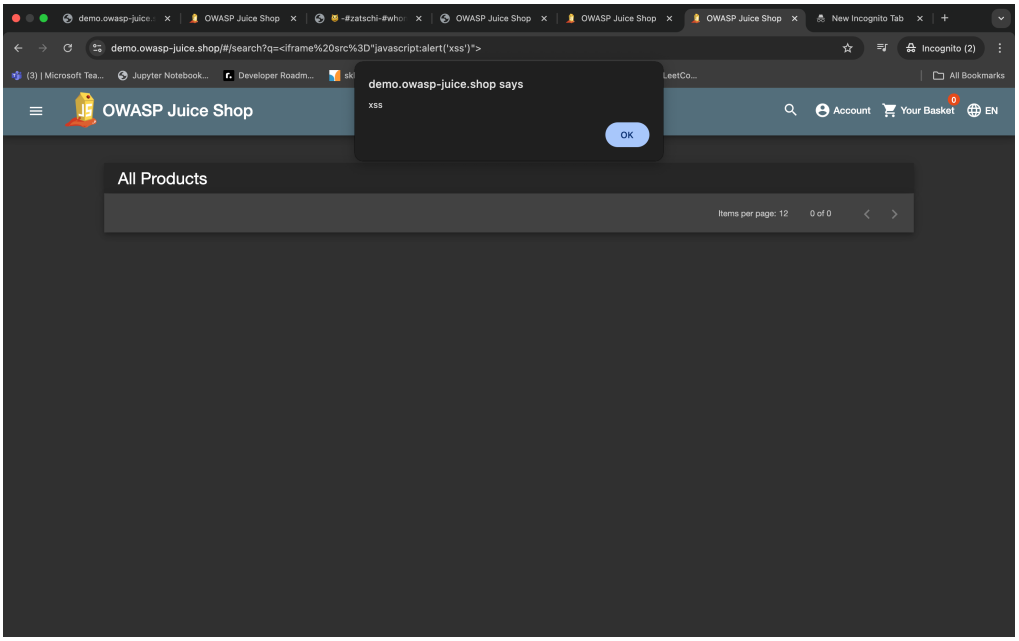


4. Navigate to http://192.168.0.103:3000/#/web3-sandbox to solve the challenge.



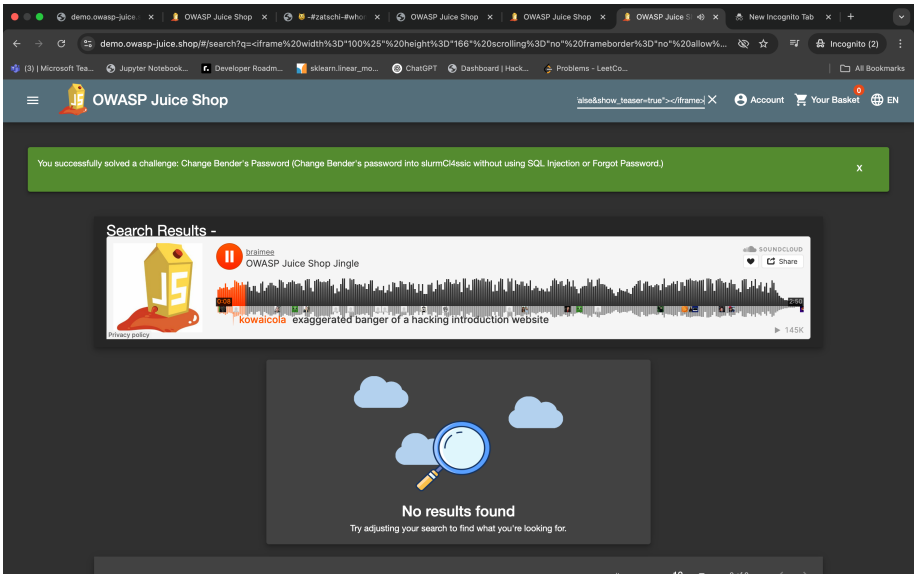
VIII.DOM XSS

1. Paste the attack string <iframe src="javascript:alert('xss')"> into the *Search…* field.
2. Hit the Enter key.
3. An alert box with the text "xss" should appear.



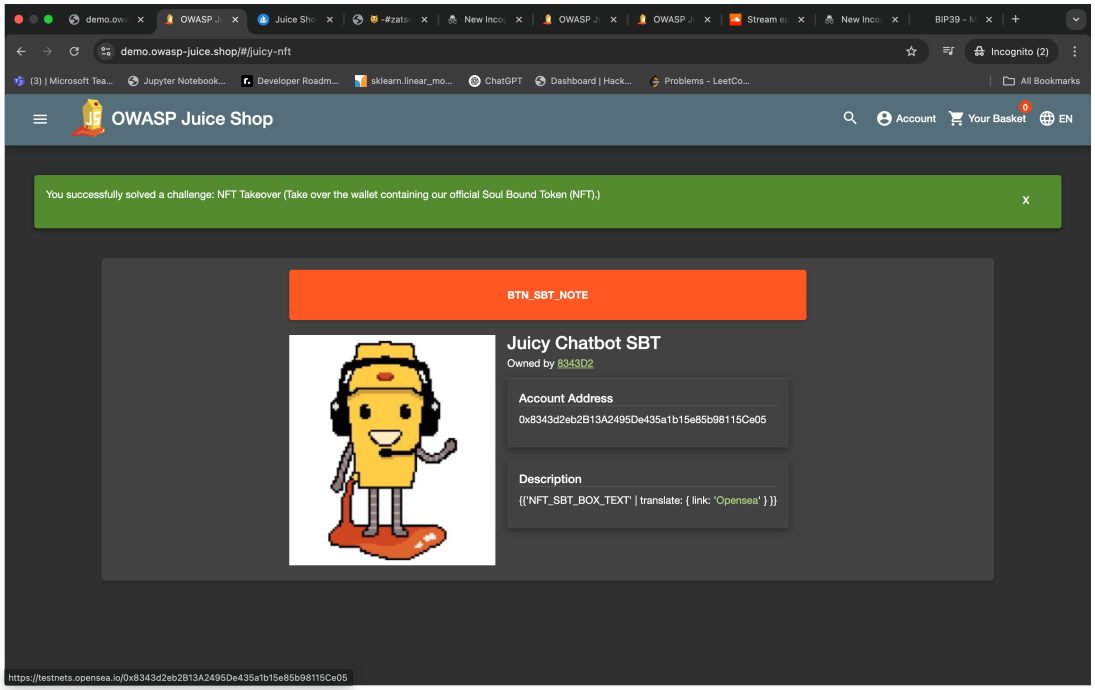
IX. Bonus Payload

1. Paste the payload <iframe width="100%" height="166" scrolling="no" frameborder="no" allow="autoplay" src="https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com/tracks/771984076&color=%23ff5500&auto_play=true&hide_related=false&show_comments=true&show_user=true&show_reposts=false&show_teaser=true"></iframe> into the *Search…* field and hit Enter



X. NFT Unlock

1. Go to the About Us section and check for the comment "Please send me the juicy chatbot NFT in my wallet at /juicy-nft".
2. Find the 12 word seedphrase of the crypto wallet in that comment "purpose betray marriage blame crunch monitor spin slide donate sport lift clutch".
3. Visit /juicy-nft in the Juice Shop App.
4. You can see an input box to enter the private keys of the wallet to access the NFT.
5. Use the seedphrase found in the comment to derive the private key of the first Ethereum wallet. You can visit https://iancoleman.io/bip39/ to get the same.
6. Enter the private key derived in the input box



Impact:
Accessing the Score Board in OWASP Juice Shop reveals a comprehensive list of security challenges embedded within the application. Each challenge corresponds to a specific vulnerability, providing insights into potential security weaknesses. While the Score Board is intentionally hidden to simulate real-world scenarios where vulnerabilities are not openly advertised, discovering it can have significant security implications.

Exposure of Vulnerabilities: The Score Board lists various security challenges, effectively disclosing the application's known vulnerabilities. If an attacker gains access to this list, they can systematically exploit these weaknesses, leading to unauthorized data access, privilege escalation, and service disruption. By knowing which vulnerabilities exist, an attacker can directly target the most critical ones, increasing the likelihood of a successful breach.

Prioritization of Attacks: With detailed information about each vulnerability's nature and difficulty, an attacker can prioritize exploits based on their skill level and potential impact. Instead of blindly testing for weaknesses, an attacker can focus on vulnerabilities that yield the highest reward, such as SQL injection, authentication bypasses, or administrative access exploits, making their attacks more efficient and damaging.

Bypassing Security Measures: Some challenges involve circumventing authentication mechanisms, exploiting access control flaws, or manipulating application logic. If an attacker understands how these challenges are designed, they can craft specific strategies to bypass security controls, compromising the application's integrity. This could lead to unauthorized admin access, modification of user data, or unauthorized financial transactions.

Reputation and Trust: If external users, stakeholders, or security researchers become aware that the application's vulnerabilities are listed and accessible—even if hidden—it can significantly damage the organization's reputation and trust. Customers and investors may question the security posture of the system, leading to loss of business, regulatory scrutiny, and reputitional harm. Public disclosure of an application's known weaknesses could also make it a prime target for cybercriminals, further increasing security risks.

Conclusion:

The unauthorized discovery of the Score Board in OWASP Juice Shop presents significant security risks by exposing a structured list of vulnerabilities within the application. This knowledge enables attackers to systematically exploit weaknesses, prioritize their attacks for maximum impact, and bypass critical security measures. The ability to gain unauthorized access, escalate privileges, and compromise sensitive data poses severe threats to the integrity of the system.

Additionally, public awareness of the application's vulnerabilities can erode trust, damage reputation, and invite targeted attacks from cybercriminals. While the Score Board is a valuable educational tool for ethical hacking and security training, its access must be strictly controlled to prevent malicious exploitation. Organizations must implement strong access restrictions, regular security audits, and continuous monitoring to safeguard against unauthorized access and mitigate potential risks. Ensuring that security vulnerabilities remain undisclosed to unauthorized users is crucial for maintaining a secure and resilient application.